Universidade Federal do Piauí – UFPI
Centro de Ciência da Natureza – CCN
Departamento de Computação – DC
Bacharelado em Ciência da Computação

Tabela Hash

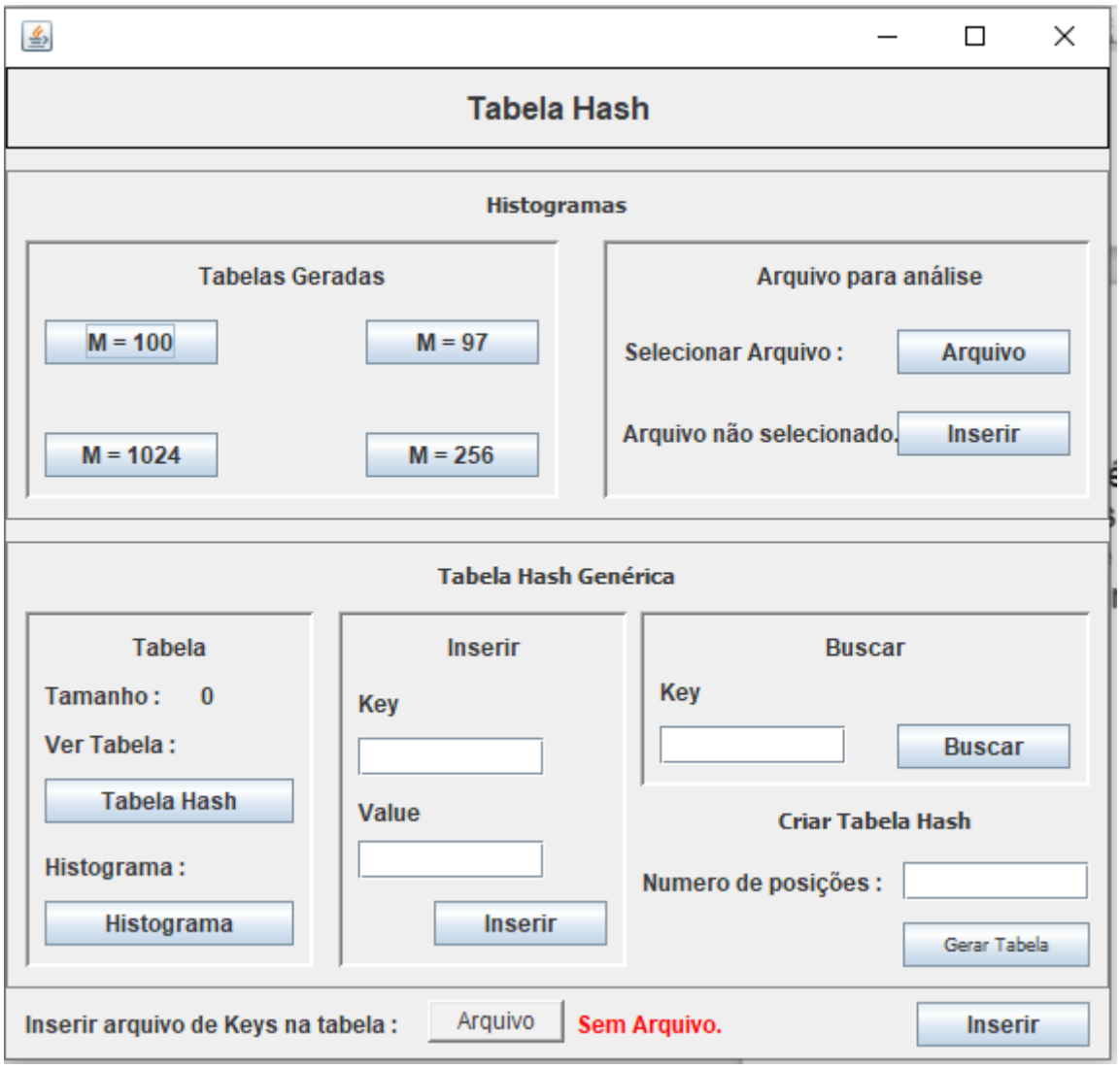
- Relatório, para a matéria de Estrutura de Dados II, sobre a aplicação criada que exemplifica e testa uma Tabela Hash com as operações de inserção e busca.

Paulo Eduardo Ramos de Araújo
Amós Lima Magalhães
Teresina – Piauí
24/06/2019

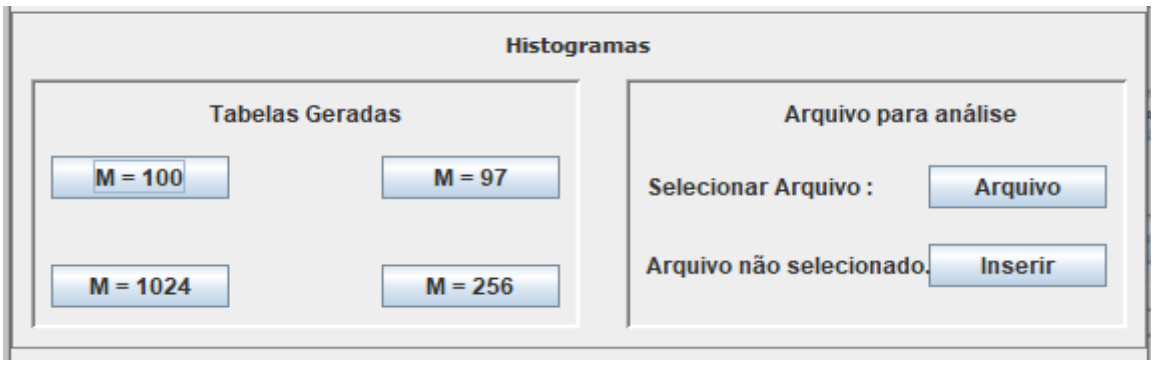
Em ciência da computação, uma tabela de dispersão (também conhecida por tabela de espalhamento ou tabela hash, do inglês hash) é uma estrutura de dados especial, que associa chaves de pesquisa a valores. Seu objetivo é, a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado. Para saber se uma tabela está balanceada temos que analisar o fator de carga C = N/M, com N igual a quantidade de itens e M o tamanho hash. Uma tabela sempre será uma tabela hash com encadeiamento se seu fator de carga for maior que 1, já que para isso a quantidade de itens deve ser maior que a quantidade(M) de slot da tabela.

Na tabela hash os métodos min(), max(), floor() e ceiling() são métodos difíceis de implementar, visto que os objetos são jogados na tabela de forma esporádica e baseada no seu conteúdo, não na ordem dos objetos, logo a busca de um objeto com maior valor ou menor valor em relação ao todo ou a outro objeto se torna difícil e sua implementação faria muitas comparações e teria um custo computacional bastante alto.

Foi desenvolvido um programa que gera histogramas para várias configurações de tabela hash, e também foi criado uma tabela genérica que é criada com as especificações do usuário.



O programa aceita inserção na tabela por meio de um arquivo de texto, este arquivo é brevemente tratado para excluir palavras repetidas e caracteres especiais.



A primeira parte do programa gera quatro histrogramas diferentes, um para cada configuração de tabela, sendo que algum deles usam função hash diferentes, ao clicar nos botões é aberto o histograma correspondente à tabela hash gerada do arquivo inserido.

As tabelas M(100), M(256) e M(1024) utilizam o hash nativo do java (object.hashCode();), enquanto a tabela M(97) usa a função hash abaixo :

```
private int hash(Key x) {
        int h = 0;
        String key = (String) x;
        for (int i=0; i<key.length(); i++)
             h = (31 * h + key.charAt(i)) % this.m;
        return h;
    }
```

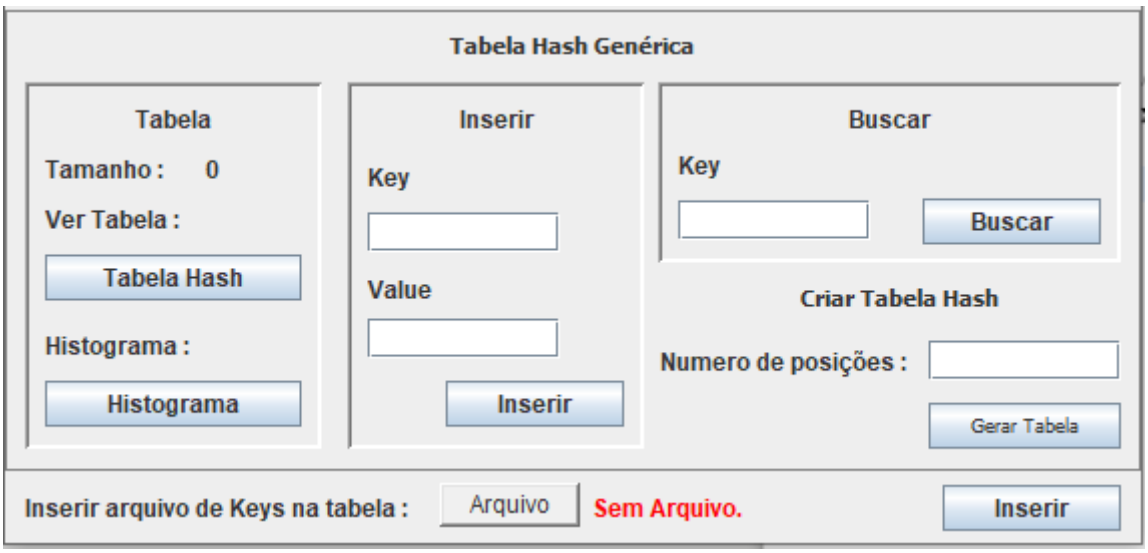
O método delete foi inserido no código das tabelas, este método faz com que o anterior ao objeto a ser excluído aponte para o próximo do objeto excluído, como exemplificado no código abaixo:

```
public class SeparateChainingHash2ST<Key, Value> {

private Node delete(Node x, Key key) {
        if (x == null) return null;
        if (key.equals(x.key)) {
            n--;
            return x.next;
        }
        x.next = delete(x.next, key);
        return x;
    }

}
```

Este método pertence a classe de uma SequencialSearchST e é chamado pela classe HashTable para o valor Hash do objeto a ser excluido.



Esta parte do programa é uma tabela hash gerada pelo usuário e usada para responder a sexta questão do trabalho, que pede a inserção da palavra E A S Y Q U E S T I O N, essa tabela funciona com a função hash exigida na questão :

```
private int hash(Key x) {
       int k = 0;
         String key = (String) x;
         for (int i=0; i<key.length(); i++)
              k = (11 * k + key.charAt(i)) % this.m;
         return k;
    }
```

Primeiramente você deve gerar a tabela com o tamanho desejado, depois pode ser feita a operação de inserção de um elemento ou inserção de um txt, o tamanho da tabela é atualizado automaticamente a cada inserção. Tem também a operação de busca e a opção de exibir na tabela o Histograma da tabela.